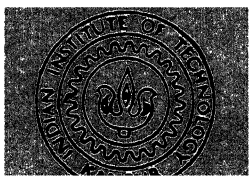
COMPUTING AN AREA-OPTIMAL CONVEX POLYGONAL STABBER OF A SET OF ISOTHETIC LINE SEGMENTS

by

CHANCHAL KUMAR



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

APRIL, 1993

Title of the Thesis

COMPUTING AN AREA-OPTIMAL CONVEX POLYGONAL STABBER OF A SET OF ISOTHETIC LINE SEGMENTS

A Thesis Submitted

in Partial Fulfillment of the Requirements

for the Degree of

Master of Technology

by

Chanchal Kumar

to the

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

April, 1993

CERTIFICATE

Certified that the work contained in the thesis entitled " *Computing an area-optimal convex polygonal stabber of a set of isothetic line segments*", by "*Chanchal Kumar*", has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Dr. Ashish Mukhopadhyay)

Department of Computer Science & Engineering,

Indian Institute of Technology,

April, 1993 Kanpur.

ABSTRACT

This thesis presents an algorithm for minimal area intersecting convex polygon for a set of isothetic line segments in R^2. The optimisation measure is area with the algorithmic time complexity of $O(n \log n)$ and linear space requirement ,which is optimal. Traversal results for simple and convex objects are also reported.

An online version of the algorithm is given.The total dynamisation of the algorithm has been successfully attempted and this is achieved in $O(\log^2 n)$ time for the most generic case, when we consider both insertion as well as deletion. But for the case when a new line segment is inserted between the extreme line segments, the procedure takes $O(\log n)$ time. Because of this optimal time requirement in a specific query range, this procedure renders itself to an optimal incremental algorithm also.

ACKNOWLEDGEMENTS

I am deeply indebted to Dr. Ashish Mukhopadhayay for his guidance and support. His dedication to work will always be a source of inspiration to me.

I will fondly remember the cooperation of Rathore and Shreesh. I also thank Alok. Tamatar. Neeraj. Rahul. Sudhir. Mohalik. Peevush. Tara. Veena. Shilpa. Rita. Sumedha. Durga. Pandey. Nema. Mishra. Vivek. SSG. Ashok. Bhusan. Golu. Das. Kholu. et al for providing me a very conducive environment. This work would have been incomplete but for the presence of a person, who chose to inspire me from a distance.

CHANCHAL KUMAR

DEDICATED TO THOSE

WHO GAVE ME THE

LEASE OF LIFE

Contents

List of Figures

Chapter 1

Introduction

1.1 What is Computational Geometry.

In the last few years, a flurry of activity can be observed in the design of algorithms for geometric problems. This can be interpreted as being caused by growing interest in the manipulation of graphical data and need of extraction of valuable information from it. It may also be caused by the fact that human intuition is often helped by using a geometric setting for practical problems. Some examples are mentioned below in the same section.

Computational Geometry as a field of Computer Science specifically deals with the subject of manapulation and analysis of graphical representation of information. The objects of interest are various geometrical entities which are subjected to several constraints and conditions the nature of which is also geometrical. It also deals with combinatorial analysis and classification of geometrical objects. Geometry being the oldest branch in mathematics, has

plenty of uses in both pure and applied mathematics. Computaional Geometry, likewise, holds a similar position in Computer Science. For example, one of the theoretical uses of Computational Geometry is the characterisation and proving of existence of geometrical objects — which is very important to many problems of other fields in Computer Science.

Computational Geometry has many practical applications also. The primary use of this field is in Computer Aided Design and Computer Aided Manufacturing. Even the study of Computer Graphics and Animation has grown out of Computational Geometry. Among other uses, Image processing, VLSI Designing specifically make use of fundamental algorithms of this field. For example, hidden line removal algorithm and convex hull algorithm are used almost everywhere and taught in the courses of very basic level in computer graphics and algorithms respectively.

1.2 Motivation for Stabbing Problem of Line Segments.

Collection of parallel line segments appear in many facets of image processing. Objects displayed on a CRT are composed of segments of parallel scan lines. Bilevel images can be stored and processed using run length encoding which represents a line segment by storing its length and one endpoint. A robot forming an image of a part moving on a conveyor belt can use structured

lighting and a linear camera array to take pictures of parallel strips of the part as it passes by. Thus line segments is a basic concept in computational geometry. In most of the applications the line segments appear parallel to each other. So we deal with parallel line segments in the present study.

In this thesis we look at the problem of stabbing — when a straight line or convex polygon can be fitted to a collection of parallel line segments so as to optimise various measures, such as Euclidean distance, area, perimeter, etc. We say that a geometrical object such as a straight line stabs a given set of geometrical objects such as convex polygons if, for this case, the line intersects every polygon in the set. The stabber as well the objects being stabbed can be taken from any class of geometrical objects. In our case the stabber is a convex polygon and the objects that are stabbed are line segments. There are two different cases of this problem — one is stabbing with the boundary of convex polygon and the other is stabbing with the interior. We look at the problem of stabbing a set of parallel line segments with the interior of convex polygons. As it can be seen in the next few lines, this problem is very easy to solve. One of the trivial solutions to this problem is a sufficient large rectangle which envelops all the line segments. To make the problem interesting we add the criterion of also optimising the area of polygon i.e. we find a polygonal stabber which has minimum area among the this class of stabbers.

This problem is generalisation of the concept of convex hull. According

to this schema of representation, a convex hull is a minimum area stabber that stabs a given set of points. The points are stabbed by the interior of the polygon so that the points that are not extreme points are also taken care of by this definition of convex hull. In fact, the algorithm of convex hull is a special case of our algorithm when all the given line segments have zero length.

A summary of relevant literature on stabbing problems that are tackled till now in computational geometry is given in the next chapter. In this thesis we enlarge the results in stabbing problems by solving the problem of computing a minimum area convex polygon that intersects a set of parallel line segments in optimal time. We say that a line segment intersects (or is stabbed by) a convex polygon if it intersects the interior or the boundary of the polygon as mentioned above. Our algorithm has the time complexity of $O(n \log n)$ and has linear space complexity. Sorting problem can be reduced to the problem of finding minimum area stabbing polygon of points on a circle. By reduction from sorting, any algorithm that outputs the stabbing polygon in clockwise order must take $\omega O(n \log n)$ steps to find a stabber of n points on a circle. Also, the space complexity of the output may be linear in input size. Thus, our algorithm is optimal in time and space.

All the algorithms are based on the solution of a new fundamental geometric optimisation problem that is of independent interest and should find application in different contexts as well.

1.3 Organisation of Thesis.

The thesis is organised as follows. In the next chapter we give a literature survey. In Chapter 3 we introduce some notations and definitions. The off-line algorithm is described and analysed in the same chapter. In Chapter 4 the on-line and dynamic version of the algorithm is described. In the last Chapter we conclude the paper and also indicate directions for further research. We look at stabbing problem for isothetic line segments in detail.

Chapter 2

Literature Survey

2.1 Transversals of Geometrical Objects.

The problem of computing a transversal of a set of ovals (an oval is a closed, bounded and convex set) is an important problem in Combinatorial Geometry. (Transversal is a synonym for stabber.) Quite a substantial amount of literature exists on the solutions to this problem and its various extensions[HDK64]. Interest in the algorithmic aspects of this problem is however quite recent. Depending on the type of the transversal and the set of ovals we get different variants of the basic problem. We mention some of the important ones that have been studied. O'Rourke gave an on-line algorithm for intersecting a set of parallel line segments by straight lines [O'R81]. Computing transversals for a set of arbitrarily oriented line segments was discussed by Edelsbrunner et al[EMP+82].

Determining line transversals of more complex objects than line segments

was studied by Edelsbrunner [Ede85]. Atallah and Bajaj gave a general technique for computing line transversals of a set of objects requiring constant size storage description by tying up the problem with that of computing Davenport-Schnizel sequences[AB87]. This result was further generalised by Srinivasaraghavan et al [SM92]. Avis and Doskas[AD87], Avis and Wenger [AW89] studied the problems of computing line transversals for lines, line segments and polyhedra in three and higher dimensional spaces.

Edelsbrunner et al [EOD81] developed a method for solving planar visibility problems that yield a procedure for computing traversals for F, a family of simple objects, in $O(n^2 \log n)$ time, where n is the cardinality of F. By simple objects, it is meant that those objects that have an $O(1)$ storage descrption and which are such that, for every pair of objects, constant time suffices to compute such basic operations as their intersections, common tangents, etc.. Given a family K of n convex cones, determining whether K admits a common traversal can be accomplished in $O(n \log n \alpha(n))$ time, where $\alpha(n)$ is the extremely slowly growing inverse Ackermann's function, with the technique of Atallah and Bajaj[AB87].

2.2 Stabbers of Line Segments.

Stabbing segments with a straight line is an important subproblem in vectorizing scanned images, computing visibility for graphical display, and finding

shortest paths for motion planning. This problem has been considered by mathematicians such as Grunbam [B.G58] and Katchalski, Lewis, and Liu [MTA86], who were studying the existence of traversals.

Goodrich and Snoeyink were probably the first to consider the a different kind of transversal than straight lines. They studied the problem of computing a convex polygon that stabs a set of parallel line segments [GS90]. One can generalise the notion of stabbing with a line to stabbing with a *convex polygon*, and the problem is attributed to Tamir [Tam87]. In such cases, the existence of the solution has to be checked and ascertained.

By defining a measure m on the class of transversal in question we get an optimisation problem. An example is that of computing the shortest line segment that intersects a set of discs. In this case, corresponding to our earlier definition, the ovals are discs, the class of transversal is the set of all line segments and the corresponding measure is the length of a line segment. Currently, there are very few results available in such kind of optimisation problem. Lyons et al studied the problem of computing a minimum perimeter convex polygon that intersects a set of isothetic line segments [LMR].A number of $O(n \log n)$ time algorithms for computing a shortest line segment that intersects a family of lines, a family of line segments and finally a family of convex polygons were given by Bhattacharya et al[BCE+91]. Jadhav et al [JMB92] gave a linear time algorithm for computing a smallest radius circle that intersects a set of convex

polygons.

In 1986, Focke[Foc86] presented an algorithm for computing the optimum perimeter inpolygon, using a finite descent method resulting from Schwarz reflection principle and co-ordinate wise descent. At a workshop, Toussaint [G.T90] posed the problem somewhat differnetly: he asked for the shortest closed path inside a simple polygon which visits every edge at least once. If the polygon is convex, then, the optimum path is, in fact, the minimum perimeter inpolygon, and [CEE+91] give a linear time algorithm which uses the reflection principle and shortest path-maps.

We look at a related problem which is a natural variation on stabbing geometric objects with a convex polygon. Rather than restrict ourselves to stabbing objects with the boundary of a polygon, we will allow the interior of the convex polygon to stab as well. In essence, we want to compute a convex polygon such that at least one point of every object is covered. In [CEE+91] a minimum covering polygon is found in $O(n)$ time for the case when the line segments are edges of a convex polygon. In [LMR] Rappaport and Meijer reported $O(n \log n)$ procedure to find the stabbing polygon with minimum perimeter. By making use of the *reflection principle* a minimum perimeter stabbing polygon for a set of isothetic line segments could be computed in $O(n \log n)$ time.

Chapter 3

Computing an area-optimal convex polygonal stabber of a set of parallel line segments

3.1 Introduction

The problem of intersecting a collection of convex objects with a common line has received considerable attention in the area of discrete and computational geometry. Such a line is known as *line traversal* or a line stabber. Algorithms for computing line stabbers include [O'R81] for computing a line stabber for a set of parallel line segments and [EMP+82] for computing a line stabber for a set of arbitrary line segments.

3.2 Definitions and Notations

We denote a line segment with end-points p and q by $\overline{pq}$. Let S denote a set of n parallel line segments. If s is a member of S then we will use the functions

top(.) and bot(.) to return the upper and lower end-points of s. Further, we will adopt the convention of writing a line-segment with its upper end-point first and lower end-point second.

The upper chain of the convex hull of the lower end-points of the line-segments in S has the property that bot(s) of each line-segment s lies on or below it. If we partially order convex chains over a given range of x-values by defining a chain to be "less than or equal" to another if at every point of the range the corresponding y-value of the former is less than or equal to the corresponding y-value of the latter, then the upper hull of the lower end-points is the "smallest" one in the given partial order to have the above property . To reflect this we denote this *l*owest *u*pward-convex *c*hain by $luc(S)$.

Similarly the lower chain of the convex hull of the upper end-points is the "largest" among all convex chains which have top(s) for each line segment s lying on or above it. We denote this *h*ighest *d*ownward-convex *c*hain by $hdc(S)$.

We assume, without loss of generality, that there is a unique leftmost line-segment $\overline{lL}$ and a unique rightmost line-segment $\overline{rR}$. Clearly, the end-points of $luc(S)$ are L and R and those of $hdc(S)$, l and r. Let $< u_1, u_2, ..., u_p >$ be the ordered set of vertices on $luc(S)$ from L to R and $< v_1, v_2, ..., v_q >$ those on $hdc(S)$ from l to r. Let P' be the convex region that consists of the points lying on or below $luc(S)$ and on or above $hdc(S)$ (Fig. 1).

If P is a convex polygon that intersects all the line-segments in S then in

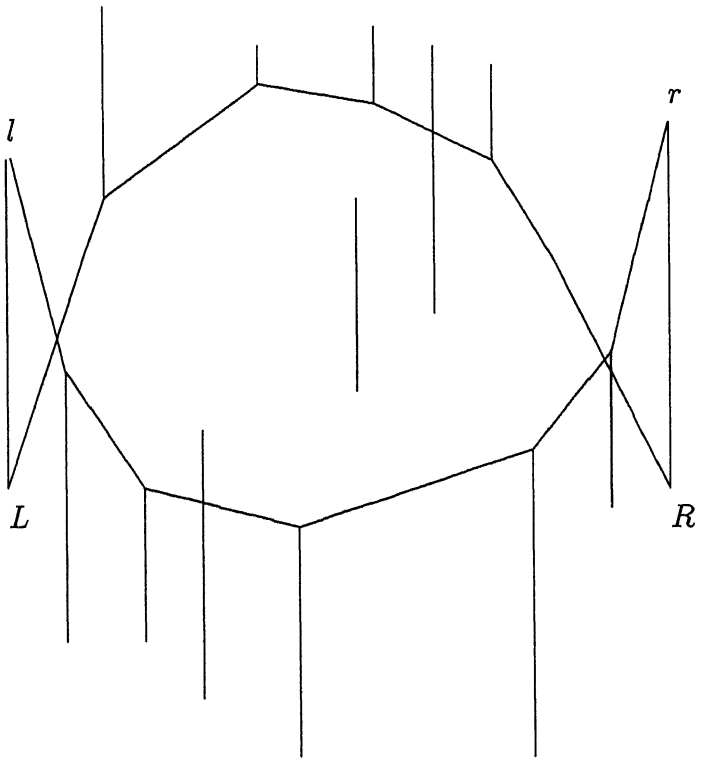


Figure 1: Convex chains $luc(S)$ and $hdc(S)$ for a set of line segments.

the range R of x-values between $\overline{lL}$ and $\overline{rR}$ the upper hull of P lies "on or above" $luc(S)$ at every value of R. Otherwise, it intersects $luc(S)$ and because of the convexities of the chains involved some u_i lies outside P. Therefore P does not intersect the corresponding line-segment - a contradiction. Similarly, its lower hull lies "on or below" $hdc(S)$ at every value of x in R. It follows that P' is contained in P. Further we can always replace P by the convex polygon that results if we truncate it by the lines supporting the leftmost and rightmost segments. It is not hard to see that we can restrict our search for an area-optimal polygon P among the class of polygons that has exactly one point in common with each of the extreme line segments.

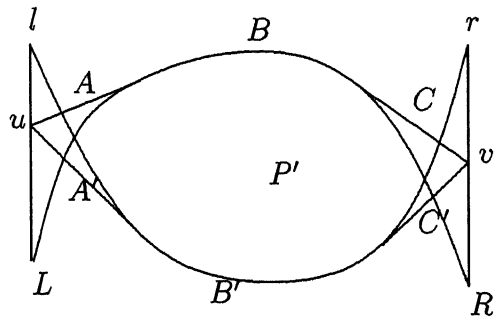


Figure 2: Parts $A(A')$, $B(B')$ and $C(C')$ of the upper(lower) chain of a convex polygon, enclosing P'.

Let u and v be the vertices of P that lie on the leftmost and rightmost line-segments respectively; $UC(P)$ and $LC(P)$ are its upper and lower chain respectively.

3.3 The Algorithm

We will first obtain a complete characterisation of the area-optimal convex polygon that intersects all the line segments in S. Our algorithm will then easily follow from this.

In the degenerate case when the leftmost and rightmost line-segments are of length zero the join of $luc(S)$ and $hdc(S)$ is the required area-optimal convex polygon. To eliminate this trivial case, we assume that these line segments are of non-zero length.

It is intuitively clear that $UC(P)$ $(LC(P))$ must hug $luc(S)$ $(hdc(S))$ as closely as possible. The problem is to express this in a quantitative way. We argue for $UC(P)$; the argument is identical for $LC(P)$.

The chain $UC(P)$ has three distinct parts - a left part A between u and the first point of contact of $UC(P)$ with $luc(S)$, a middle part B that is common with $luc(S)$ and a right part C between the last point of contact of $UC(P)$ with $luc(S)$ and v (Fig. 2).

It is obvious that $A(B)$ cannot have any vertex of UC(P) lying strictly between $u(v)$ and the first(last) point of contact between $UC(P)$ and $luc(S)$. Otherwise P would not be of minimum area. Therefore the parts A and B are straight-line segments. If we extend these, each is tangent to $luc(S)$ along an entire edge or at a vertex only.

Let A', B' and C' be the corresponding parts of LC(P). The following lemmas provide a necessary characterisation of P.

Lemma 3.1 *If P is a convex polygon of minimum area that intersects all the line segments in S, then either A (respectively B), when extended, touches $luc(S)$ along an edge or A'(respectively B'), when extended, touches $hdc(S)$ along an edge.*

Proof: Suppose that neither A' nor A, when extended, touches its respective chain along an edge. Let x and y be the contact points (Fig. 3) If $\overline{xy}$ is vertical then we can move either $\overline{ux}$ or $\overline{uy}$ to satisfy the condition of the theorem without changing the area of the polygon P. If $\overline{xy}$ is not vertical let it be skewed as in the figure above. Clearly if we move $\overline{ux}$ by moving u towards top($\overline{lL}$) then we get a polygon of smaller area than P, contradicting its optimality. A similar

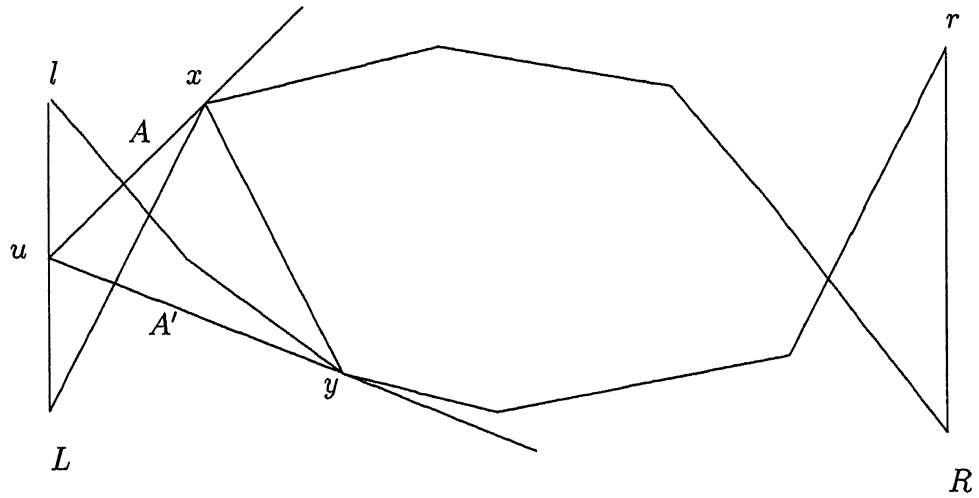


Figure 3: Edges A and A', touching P' at x and y

argument holds if $\overline{xy}$ is skewed towards top($\overline{lL}$). Thus one of A or A' must satisfy the condition of the theorem. ∎

If one of A or A', when extended, touches along an edge and the other at a vertex, the edge and the vertex are related as in the lemma below.

Lemma 3.2 *If A when extended touches the edge $\overline{xy}$ (vertex z) of $hdc(S)$ and A' when extended touches the vertex z (edge $\overline{xy}$) of $luc(S)$ then the vertical line through z intersects $\overline{xy}$.*

Proof: Suppose x and y are to the right of the vertical line through z (Fig. 4). Given the skew of $\overline{xy}$ relative to $\overline{lL}$, by moving the vertex u of the triangle $\triangle uxz$ towards top($\overline{lL}$) we reduce the area of P. This contradicts that P is of minimum area. A similar argument holds when x and y are to the left of the vertical line through z. Thus when P is of minimum area the vertical through z intersects $\overline{xy}$. ∎

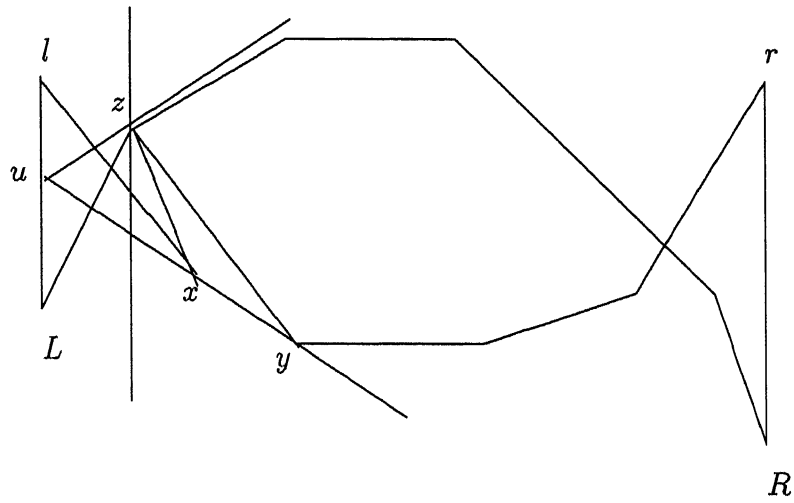


Figure 4: Edge of contact $\overline{xy}$, lying to the right of contact point z.

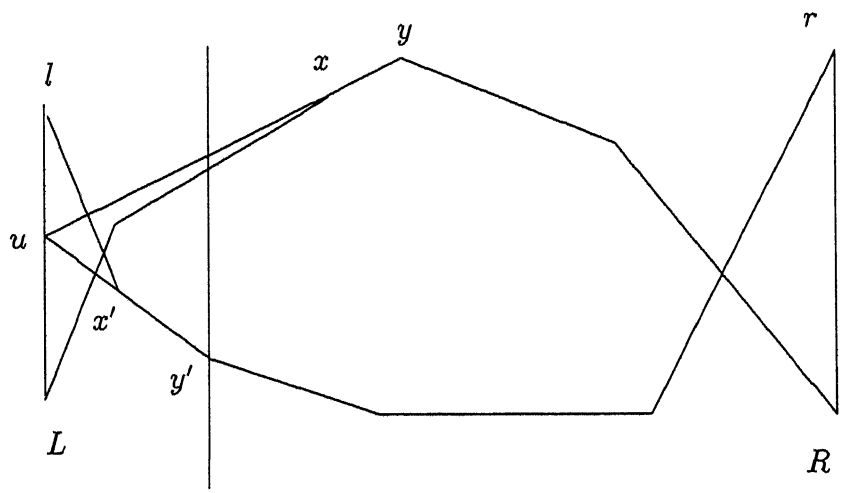


Figure 5: Edge of contact $\overline{xy}$, lying to the right of edge of contact $\overline{x'y'}$.

Lemma 3.3 *If A when extended touches the edge $\overline{uv}$ of $hdc(S)$ and A' when extended touches the edge $\overline{u'v'}$ of $luc(S)$ then either the vertical line through u intersects $\overline{u'v'}$ or the vertical line through u' intersects $\overline{uv}$.*

Proof: The proof goes along the lines of the proof of **Lemma 2.** The details can be easily filled in by a careful study of Fig 5. ∎

Lemmas 2 and **3** above hold when we replace A and A' by B and B' respectively.

The above characterisation of the area-optimal convex polygon that intersects the line segments of S is necessary. We show that such a polygon is unique, except in the case when $luc(S)$ and $hdc(S)$ are mirror images of each other, which essentially proves the sufficiency of the above characterisation. Let vl be the vertical line of **Lemma 2** or **3**. If there is a another convex polygon enclosing P' for which one of **Lemma 2** or **3** hold then the corresponding vertical line lies strictly to the left or right of the vertical line vl. Suppose that it lies to the right of vl. Then the parts A and A' do not join at a point on $\overline{lL}$, but to the left of it. Similarly if it lies to the left of vl then the parts A and A' intersect to the right of $\overline{lL}$. Thus **Lemmas 2** and **3** constitute a complete characterisation of the area-optimal convex polygon which intersects the line segments in S.

To compute A and A' we consider an edge of $luc(S)$ or $hdc(S)$ and extend it. If it intersects the leftmost segment, we draw a tangent from the point of

intersection to the other chain, and check if any of **Lemmas 2** or **3** applies. If not, we repeat until such an edge is found. Parts B and B' are computed similarly.

We can now formally state the algorithm to compute the parts A and A'.

Algorithm `MinPolyStabber(`S, A, A'`)`

Step 1. `Compute the upper hull` $luc(S)$ `of the points bot(`s`) and the lower hull` $hdc(S)$ `of the points top(`s`).`

Step 2. `If all edges on` $luc(S)$ `are marked then`

`go to` **Step 3**

`else`

`choose an unmarked edge` e `on` $luc(S)$`;`

`if the supporting line of` e `does not intersect` $\overline{lL}$ `then`

`mark` e `and go to` **Step 2**

`else`

`draw a tangent to` $hdc(S)$ `from the point of intersection;`

`if` **Lemma 2** `or` **Lemma 3** `applies to the configuration of tangents so obtained then`

`return` A, A' `and quit`

`else`

`mark` e `and go to` **Step 2**.

Step 3. Repeat **Step 2** with $hdc(S)$ replaced by $luc(S)$ and vice versa, returning B, B' instead of A, A'.

3.4 Analysis of the Algorithm

The complexity of Step 1 is bounded above by $O(n \log n)$. Since it requires time logarithmic in the size of a chain to compute a tangent to it from a given point, the complexity of Step 2 is bounded above by $O(n \log n)$ also. Same goes for Step 3. Therefore the time complexity of the algorithm to compute parts A and A' is $O(n \log n)$. Since B and B' can be computed in the same time complexity, the time complexity of the algorithm is $O(n \log n)$.

Chapter 4

Online Algorithm for Polygonal Stabber

4.1 Introduction

This chapter deals with the online version of the minimum area covering polygon for set, S, of isothetic line segments.

The online version of the problem has been attempted and a general approach has been presented, with a complexity of $O(\log^2 n)$ and linear space requirement. Here, certain special properties of convex polygons have been exploited to give an $O(\log n)$ step algorithm for the above case.

4.2 Geometric preliminaries

The idea mooted in the last chapter was to find a configuration of the convex polygon such that it satisfies the Lemma 2 and Lemma 3 of the previous chapter.Lemma 2 states that A when extended touches the edge $\overline{xy}$ of $hdc(S)$ and

A' when extended touches the vertex z of $luc(S)$ then the vertical line through z intersects $\overline{xy}$. Following this, the approach should be to find this vertical line through z, which we know exists for sure. So the following section will deal with the problem of determining such a vertical line.

Let $top(s)$ and $bot(s)$ give the upper and lower endpoints of the line segment s belonging to S. We visualise two cases of the new line being inserted in the domain space. The given line segment $\overline{gG}$ lies in between the extreme line segments $\overline{rR}$ and $\overline{lL}$ in the first case and in the second case, the line segment $\overline{gG}$ lies to the right of $\overline{rR}$ (or to the left of $\overline{lL}$).

Case 1

When the given segment $\overline{gG}$ lies to the right of the rightmost line segment $\overline{rR}$ i.e.

$x - value(\overline{gG}) \geq x - value(\overline{rR})$.

The case of

$x - value(\overline{gG}) \leq x - value(\overline{lL})$

can be treated in a likewise manner.

Introducing the lower endpoint of the segment $\overline{gG}$ i.e. $bot(\overline{gG})$ will change the upper chain luc and atmost two new edges can possibly be introduced (after deleting some of the old edges). Refer to Fig 6 .

Let the new chain be luc'. Mathematically,

$luc' = conv.hull\{luc \cup bot(\overline{gG})\}$

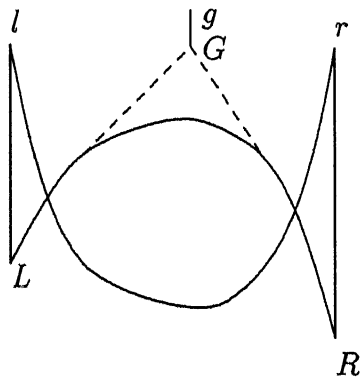


Figure 6: Updated convex chains $luc(S)$ and $hdc(S)$ on introduction of a new line segment $\overline{gG}$

Similarly, we have a relation for the lower chain:

$hdc' = conv.hull\{hdc \cup top(\overline{gG})\}$

If we maintain the insert only tree data structure, then both these operations or changes can be effected in $O(\log n)$ time each. Once we have the new upper and lower chains, the new minimum area convex polygon can be found in $O(\log^2 n)$ steps. The algorithm proceeds as follows:

Step 1: Pick a vertex x on the upper chain luc'.

Step 2: Extend the corresponding edge $\overline{xy}$ ϵ luc' to the rightmost line segment $\overline{rR}$ and mark the event point u on $\overline{rR}$.

Step 3: From the point u draw the tangent to hdc' at z.(Fig. 7).

Step 4: If segment $\overline{xz}$ (after the points are joined) is skewed and is inclined towards the segment $\overline{rR}$ (as shown in the figure), then, we move towards $\overline{rR}$ on the upper chain luc' i.e choose new vertex point x' such that

$x - value(x') > x - value(x)$.

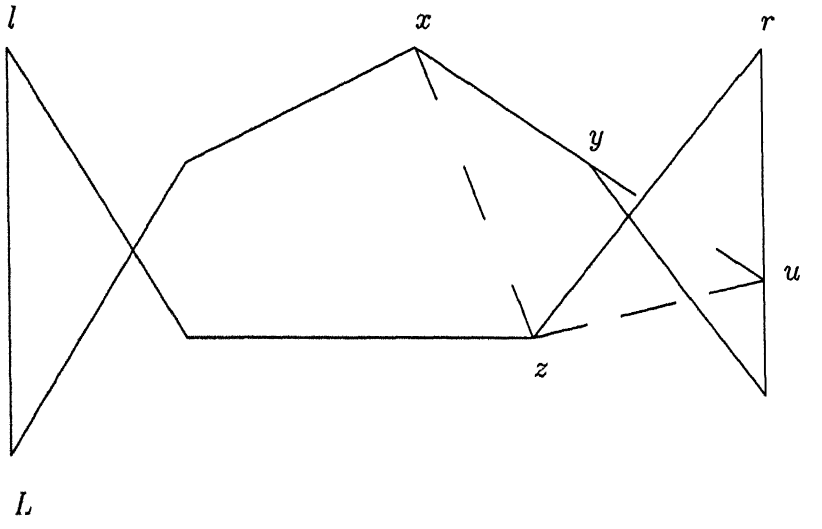


Figure 7: Movement on luc' depending on the skew of xz

If $\overline{xz}$ segment is skewed otherwise , then we move in the opposite direction. These movements are dictated by the principle of calculus, which indicates least area for vertical configuration of the line joining the vertex on the luc' and the corresponding tangent point on the hdc' (as shown in the last chapter). If $\overline{xz}$ segment is vertical, i.e. part of the translate of the Y-axis, then the optimal area $\triangle xuz$ has been reached. This triangular region gives the part C and C' of the optimal area convex polygon and it is merged with the other parts and reported alongwith the convex chain.

4.2.1 Complexity Analysis

It is easy to see that Step 1 and Step 2 take unit time. Step 3 can be done by any canonical algorithm for finding the tangent to the upper and lower chains. This can be accomplished in $O(\log n)$ time.

Step 4 gives an indication of the number of times Step 3 has to be run. This stage can be implemented by a binary search like procedure or any randomised version of such techniques, and the job can be easily done in worst case complexity of $O(\log n)$. This can be explained as follows. We pick one vertex on the upper chain and compute the corresponding tangent point on the lower chain and observe the slope of the line joining them with respect to the rightmost segment $\overline{rR}$. If this condition permits movement to the right on the upper chain, then this eliminates half the search space that lies to the left of the vertex on the upper chain. This is so because the tangent segment rotates in the planer space monotonically in clockwise or anticlockwise direction. Such half-reduction in search space is typical of logarithmic algorithms.

Step 3 and Step 4 coupled together give $O(\log^2 n)$ algorithm for finding the minimum area convex polygon covering $\overline{gG}$ and the original set S.

Case 2

When the given line segment lies in between the extreme segments, i.e.

$x - value(\overline{lL}) \leq x - value(\overline{gG}) \leq x - value(\overline{rR})$.

If the given line segment $\overline{gG}$ lies above the upper chain, then only the upper chain *luc* will change and the lower chain will remain unchanged. If the given line segment $\overline{gG}$ lies below the lower chain, then only the lower chain *hdc* will change and the upper chain will remain unchanged. If the line segment intersects any of the upper or lower chains, then, the optimality does

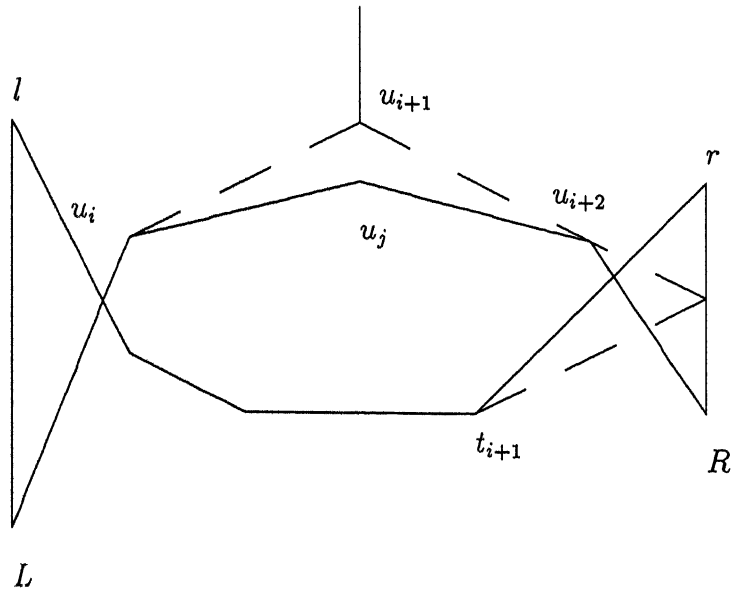


Figure 8: When the optimality of the solution changes on the introduction of a new line segment

not change and the current minimum polygon continues to be the modified optimal polygon.

From the Fig. 8, it is clear that the introduction of a new segment can potentially introduce two new edges on the upper chain (similar is the case with the lower chain). Let the two new edges be, $\overline{u_i u_{i+1}}$ and $\overline{u_{i+1} u_{i+2}}$. The analogous situation on the lower chain can be treated the same way.

We keep with a vertex u_m belonging to luc' (on upper chain), a vertex on hdc', which is tangent to the extended event point on the rightmost segment $\overline{rR}$ from u_m-edge. This information helps us in deciding the vertex or the grazing edge corresponding to the global minima position. Let the vertex corresponding to the *optimum position* be u_j on luc'. Three situations can arise with regard to the position of the vertex u_j.

Case (i)

If

$x - value(u_j) < x - value(u_i)$

i.e. the vertex u_j lies to the left of the vertex u_i. In this case the visibility relations for the vertices to the left of the point u_i on luc' or to the right of the point u_{i+2} does not change. In other words, the tangent segments corresponding to the points u_i on luc' will remain skew and will not be optimal. This implies that the global minimum does not change.

Case (ii)

If

$x - value(u_j) > x - value(u_{2+i})$

i.e. the vertex u_j lies to the right of the vertex u_i. This condition is no different from the previous one and hence, the global optimum remains unmodified. We output the same old polygon as the required new minimal area polygon.

Case (iii)

If

$x - value(u_i) <= x - value(u_j) <= x - value(u_{i+2})$

then, the optimality suffers a change. The algorithms and the procedures described in the previous chapter show that this condition implies that the

tangent segments corresponding to the vertices u_{i-1} and u_{i+3} on luc' are oppositely inclined towards $\overline{rR}$ (this means that the two segments have positive and negative slopes in the XY plane) . Hence, the optimal condition lies somewhere in between. We now extend the edges corresponding to the point u_{i-k-1} on luc' and mark the event points on the rightmost segment $\overline{rR}$ for k=0,1,2,3,4. Then, the corresponding tangent points t_{i-k-1} on hdc' are found. The lines joining these two points on luc' and hdc', respectively, are found (called tangent segments).

Exhaustive search of the slope of these lines $\overline{u_{i-k-1}t_{i-k-1}}$ gives the minimal condition (the slope of the line will then, be infinity in the XY plane) and this is the global minima as well. Report this as the new optimum convex polygon. This step can be performed in $O(\log n)$ time.

Earlier steps could be efficiently implemented if some amount of post-processing is done. If v_i is the vertex/tangent point on the lower chain corresponding to the vertex u_i on the upper chain and v_j is that corresponding to u_{i+2}, then, for all vertices v_k on the lower chain such that

$x - value(v_i) < x - value(v_k) < x - value(v_j)$

holds, the corresponding tangent points are marked u_{i+1}. The overall process can take amortized linear time, if the process is repeated for all the vertices and this can help complete the last step i.e, Step 4 in constant time.

The dynamisation process crucially hinges on the fact that the minimum

area polygon has the two tangent segments parallel to the Y co-ordinate axis and that any slanting segment (as part of the rightmost or the leftmost triangles) can not give the minimum area. The complexity of $O(\log n)$ is achieved because of the existence of algorithms of same complexity for the maintenance of convex hull for insertion only case.

This case easily gives us the method of finding an incremental algorithm, which is optimal as the procedure has to be repeated $O(n)$ number of times and this provides an $O(n \log n)$ complexity procedure.

Chapter 5

Dynamic Convex Polygonal Stabber Algorithm

5.1 Introduction

In many applications, one needs dynamic convex hulls, where points can be inserted or deleted from the set. The best bound currently known for the dynamic convex hull is due to Overmars et al [OL81], who show that a convex hull can be maintained at a worst case cost of $O(\log^2 n)$ per insertion or deletion. Better results are possible, however, if there are only insertions or only deletions. In the case of insertions, an algorithm due to Preparata [F.P79] can insert a new point and update the convex hull in worst case time of $O(\log n)$. The case of deletions is little more complicated, but it is possible to process a sequence of n deletions in $O(\log n)$ amortized time per deletions. The same result is also implicit, though not fully worked out, in the convex layers algorithm of Chazelle[B.C85].

In this chapter the dynamisation of the problem of stabbing a set of parallel line segments by a convex polygon has been attempted. A general approach has been presented, with a complexity of $O(\log^2 n)$ and linear space requirement. This means that the algorithm takes care of both the cases of deletion and insertion of a line segment in the plane of R^2. This algorithmic complexity turns out to be optimal when the new line segment happens to lie between the extreme line segments as explained in the last chapter. The bound achieved is largerly due to the advanced results obtained in linear programming and the efficiency of the existing algorithms for convex hulls.

5.2 Geometric preliminaries

We will follow the notations and concepts developed in the previous chapters i.e. *luc* and *hdc* will represent the lowest upper chain and highest downward chain, respectively. The new line segment introduced is $\overline{gG}$. We argue for finding the optimal polygon part C and C' incident on $\overline{rR}$ and the argumentis identical for the part incident on $\overline{lL}$.

5.3 Totally dynamic version

Here, the data structure used in maintaining dynamically the convex hull of planar set of points is exploited to our benefit. The details can be found in Overmars and Leeuwan [OL81]. To achieve this, a sophisticated tree data

structure is used.

For the relevant figures, refer to Fig. 6, Fig. 7 and Fig. 8 given in the last chapter. The algorithmic steps can be summarised as follows and the correctness of the algorithm is evident from the steps:

Step 1: In $O(\log^2 n)$ time construct the new upper chain and the lower chains (i.e. *luc* and *hdc*) with the introduction or deletion of a line segment. This is a general methodology and the applies to the line segment equally well whether it lies in between the extreme segments or beyond them.

Step 2: For vertices lying on the *luc'* and *hdc'*, draw tangents to *hdc'* and *luc'*, respectively after marking the event points on the rightmost line segments. The events points are marked by taking the intersection of the rightmost line segment $\overline{rR}$ and the extended edge u_iu_{i+1} of *luc* or *hdc* , as the case may be. Computing tangent to a convex chain takes $O(\log n)$ time.

Step 3: As the slope of the tangent segment goes from positive to negative or vice-versa, the area covered by the convex polygon decreases to a minimum and then, again increases, that is, there is a minimum position for the tangent segment. Hence, for the vertices belonging to the upper chain, implement a binsearch like procedure to find the vertex corresponding to the global minima. Thus, Step 2 is carried out for $O(\log n)$ number of vertices.

Step 4: Report the minima thus found and the proofs and the lemmas outlined in the last chapter ensures that this is actually the global minimum

area polygon.

5.3.1 Complexity Analysis

Updating the convex chains *luc* and *hdc* takes $O(\log^2 n)$ time, which is the time taken by Step 1. Then, drawing the tangents to a convex chain is accomplished in $O(\log n)$ steps. Step 3 is to be run $O(\log n)$ number of times as the monotonically changing tangent segment allows us to discard the half search space in every iteration. So, the number of time the Step 3 has to be repeated is logarithmic. This is a general search result which can be applied even in the case of algorithm described in Chapter 2. But, there as the computation of convex hull will in any case consume $O(n \log n)$ time, there is no apparent gain in time complexity. But, here this result decreases our overall complexity. Now, it is easy to see that the algorithm given takes $O(\log^2 n)$ time and this is the best bound at the moment. This is so because finding an algorithm to maintain the convex hull for planar set of points in less than $O(\log^2 n)$ time is still an open problem.

Chapter 6

Conclusion

This thesis reports a computational study of traversal problems . We have presented an $O(n \log n)$ algorithm for computing a minimal area convex polygonal disk that stabs a set of parallel line segments. The algorithm runs in $O(n \log n)$ time and $O(n)$ space, which is optimal.

Then the online version of the problem has been solved and a general approach of complexity $O(\log^2 n)$ is given, which becomes optimal in specific query range. The efficiency of the algorithms is (to a great deal) due to the efficiency of existing algorithms for convex hull problems and the low dimensional linear programming. This includes the best bound for the dynamic convex hull problem due to Overmars & van Leeuwen [OL81], who showed that a convex hull could be maintained at a worst-case cost of $O(\log^2 n)$ per insertion or deletion.

6.1 Incremental algorithm

The online algorithm easily lends itself to the *incremental algorithm* for computing minimal area convex polygon for a set S of parallel line segments. The information about the rightmost and leftmost line segments are kept while preprocessing the set S and this is the initial set of line segments to start with. This takes linear time. Then, the line segments are selected at random and the online version applied to get the modified optimal cover. Knowledge of the extreme line segments $\overline{rR}$ and $\overline{lL}$ means that the $O(\log n)$ case of the algorithm has to be applied (when the new line segment to be inserted lies between the extreme line segments). Insertion of n line segments will give an incremental version of $O(n \log n)$ complexity.

6.2 Stabber for a set of orthogonal line segments

Recent results due to Edelsbrunner substantiate the thesis that a number of problems are considerably less complex in an orthogonal environment(i.e. with axis-parallel objects only) as opposed to a non-orthogonal environment(i.e. with objects of arbitrary directions). Given a set of orthogonal line segments in R^2, we apply the concepts and the sophisticated tools of the computational geometry used in the previous chapters, and point out the limitations thereof.

6.2.1 Geometric Preliminaries

Let S be a set of n isothetic line segments in the plane, that is, the lines are parallel to the horizontal and vertical coordinate axes. Let $V = \{s_1, \ldots, s_z\}$ be the set of vertical line segments and $H = \{s_{z+1}, \ldots, s_n\}$ be the set of horizontal line segments. A polygon stabber of S is a simple polygon that intersects at least one point of each line segments in S with its boundary or its interior. This polygon is required to have minimal area and the convex nature as well.

We define the vertices of a polygon of P as $p_0, \ldots, p_k$ such that $p_0 = p_k$ and the vertex p_{i+1} follows vertex p_i in the counter clockwise walk around $P, i = 0, 1, ..., k-1$. Given a polygon P, denote the boundary of the polygon P as $bd(P)$. It is easy to see that the minimum area stabbing polygon is convex. This fact can be stated as:

Lemma 6.1 *Every minimum polygon stabber of a set of line segments is convex.*

Proof: By contradiction. ∎

The top and the bottom endpoints of the vertical line segment $s \in V$ are given by $t(s)$ and $b(s)$, respectively. Similarly, the left and the right endpoints of a horizontal segment are given by $l(s)$ and $r(s)$, respectively. Let $y(s)$ and $x(s)$ be the y-value of the horizontal line segment $s \in H$ and the x-value of the vertical line segment $s \in V$, respectively. We define the two *extreme* y-values and the two *extreme* x-values of S.

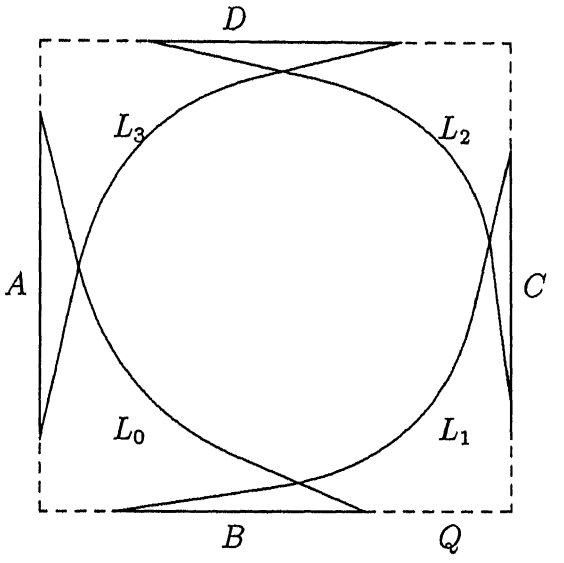


Figure 9: Closed isothetic rectangle Q through E_L,E_B,E_R, E_T

* Let the two extreme y-values be

$E_T = max\{\{maxy(s)|s \in H\}, max\{y(b(s))|s \in V\}\}$, and

$E_B = min\{\{miny(s)|s \in H\}, min\{y(t(s))|s \in V\}\}$.

* Let the two extreme x-values be

$E_R = max\{\{maxx(s)|s \in V\}, max\{x(l(s))|s \in H\}\}$, and

$E_L = min\{\{minx(s)|s \in V\}, min\{x(r(s))|s \in H\}\}$.

We define Q as the closed isothetic rectangle through E_L, E_B, E_R, and E_T. Fig. 9 shows a set of line segments and the rectangle Q. Denote the left, bottom, right, and top edges of Q as A, B, C and D, respectively.

Lemma 6.2 *A minimum area stabbing polygon P of S is entirely contained in Q.*

Proof:

Suppose P is a minimum stabbing polygon not contained in Q. We can characterize the boundary of P by a sequence of points $(R_1, a_1, O_1, b_1, R_2, a_2, \ldots)$ where each R denotes a contiguous subsequence of vertices of P contained in Q, each O denotes a contiguous sequence of vertices of P outside Q, and each a_i and b_i denote an intersection point of P and Q.

For each subsequence O in P we construct a subsequence O' in the following way. Take orthogonal projection of points to the left of Q onto the left edge and similar projections onto the three other sides.

A vertex in a corner region is replaced by the corresponding corner of Q. Represent P' as $(R_1, a_1, O_1', b_1, R_2, \ldots)$. It can be easily shown that P' is also a stabbing polygon and is contained in Q. By similar arguments, P' is found to have lesser area than P. This proves the Lemma.

∎

Lemma 6.3 *Every minimum area polygon stabber of S passes through the segments that define E_R, E_L, E_B and E_T; i.e. every optimal polygon has a vertex on the edges A, B, C and D of Q.*

Proof: Follows from Lemma 6.2. ∎

By lemma only those parts of the segments are to be considered that are in $S \cap Q$. For a point s in S we define $t(s) = b(s) = l(s) = r(s) = s$.

Define the following:

X_0 is the set of right and top endpoints of S.

X_1 is the set of left and top endpoints of S.

X_2 is the set of left and bottom endpoints of S.

X_3 is the set of right and bottom endpoints of S.

$CH(X_i)$ is the convex hull of $X_i, i = 0, 1, 2, 3$.

Let us define the following:

L_0 is the part of $CH(X_0)$ from A to B.

L_1 is the part of $CH(X_1)$ from B to C.

L_2 is the part of $CH(X_2)$ from C to D.

L_3 is the part of $CH(X_3)$ from D to A.

Also, R_0 is the area between L_0 and Q. R_1, R_2 and R_3 are analogously defined. Finding the minimal convex polygonal stabber means a convex polygon with the sides lying in the regions $R_i, i = 0, 1, 2, 3$.

Observations: Intersection of the four convex chains gives a quadruple wrapper which needs to be covered by all the stabbers. All segments have some part of it inside the wrapper. Finding four minimum area triangles on the four orthogonal sides independently may not, in general, give a convex polygon when the intersection of these triangles with the wrapper is taken. This is so, if the tangent segments for the four different sides are intersecting. In case, they are not intersecting, then the join of these portions give the optimal area polygon. But the join of even an intersecting wrappers may give a convex polygon to start with because this takes care of some trivial and degenerate cases.

The globally optimal polygon will have four anchor points on the four extreme line segments, resectively perpendicular to each other. It is observed that these anchor points can move independent of each other and even if one anchor point is fixed, the movement of remaining three anchor points will give a continous area function. So, discrete movements on the extreme segments will not always optimize the area function. In the case of isothetic line segments, it was observed that a movement to the left on the *luc* for tangent-segment resulted in a movement to right of the corresponding vertex point on the *hdc* because the tangent segment is rotating monotonically in R^2 space. It is also seen that the wrapper chains L_i for i=0,..,4 do not bear any relation with each other, unlike the case of isothetic line segments. Even the monotone condition [DR76] can not be applied as the geometric measure to be optimised is area and not the Euclidean distance. To begin with, one could independently compute the minimum area triangular regions with one vertex incident on each of the four extreme line segments. Take the convex hull of these regions with the quadruple wrapper and let the polygonal structure formed be W. Now if one tries to minimize the area of this polygon, one method could be to move to the north-east corner of the rectangular window. For this, an equation needs to be solved to get to the minimum and for this the center point needs to be fixed. Hence, for each step one equation with different center point {which is computed by the intersection of the parallel lines from the other two points

incident on the extreme segments} has to be solved. This procedure has to be repeated for all the four corner points. If there is a finite number of optimal area covering polygons, scanning all these areas will give the required polygon P.

Thus, the above procedure intuitively gives an approximately minimal area as the attempt has been to hug the quadruple as close as possible. But the problem of finding a global minima in this context needs some additional theorem proving.

The natural problem to tackle, then, is to compute a minimum area convex polygon that stabs a set of orthogonal line segments, with the ultimate goal of solving this problem for an arbitrary set of line segments - this is a simplification of Tamir's original problem, which is still open [Tam87]. Extension of these problems in three dimension is also of interest.

Finding a randomised algorithm for this stabbing problem is an interesting exercise. With the present sophistication of randomization techniques, like Polling, [JS89] it is not possible, *in general*, to solve an optimisation problem. The trick is to seek the characterisation of the solution and manifest this in the query itself as a parameter i.e. impose this characterisation as a weak condition on the solution space where the random sampling is performed [CS89].

Then, it would be a nice effort to modify these algorithms, when *only the boundary* (and not the interior) of the convex polygon is allowed to intersect

the line segments. The problem is still open if the optimisation measure is changed to something other than area and perimeter. And, it would indeed be challenging to find the optimal algorithms for the generic dynamic problem and to seek the lower bound possible for deterministic as well as randomised algorithm.

Bibliography

[AB87] M. Atallah and C. Bajaj. Efficient algorithms for common transversals. *Information Processing Letters*, 25:87–91, 1987.

[AD87] D. Avis and M. Doskas. Algorithms for high dimensional stabbing problems. *Discrete Applied mathemtics*, 27:39–48, 1987.

[AW89] D. Avis and R. Wenger. Polyhedral line traversals in space. *Discrete and Computational Geometry*, 4, 1989.

[B.C85] B.Chazelle. On the convex layers of a planar set convex hulls. *IEEE Transactions on Information Theory*, 4:509–517, 1985.

[BCE+91] B. K. Bhattacharya, J. Czyzowicz, P. Egyed, G. Toussaint, I. Stojmenovic, and J. Urrutia. Computing shortest transversals of set. In *Proc. of the Seventh Annual ACM Symp. on Computational Geometry*, pages 71–80, 1991.

[B.G58] B.Grunbaum. On common traversals. *Arch. Math.*, 9:465–469, 1958.

[CEE+91] J. Czyzowicz, P. Eyed, H. Everett, D. Rappaport, T. Shermer, D. Souvaine, G. Toussaint, and J.Urrutia. The aquarium keeper's problem. In *The 2nd Symposium on Algorithms and Data Structures*, January 1991.

[CS89] K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry ii. *Discrete and Computational Geometry*, 4:387–421, 1989.

[DR76] D.Dobkin and R.J.Lipton. Multidimensional searching problems. *Siam Journal of Computing*, 5:181–186, 1976.

[Ede85] H. Edelsbrunner. Finding transversals for sets of simple geometric figures. *Theoretical Computer Science*, 35:31–45, 1985.

[EMP+82] H. Edelsbrunner, H. A. Maurer, F. P. Preparata, A. L. Rosenberg, E. Welzl, and D. Wood. Stabbing line segments. *BIT*, 22:274–281, 1982.

[EOD81] H. Edelsbrunner, M.H. Overmars, and D.Wood. Graphics in flatland: a case study. Technical Report F79, Technical University of Graz, Tech. Report, 1981.

[Foc86] J. Focke. A finite descent method for STEINER's problem of in-polygons with minimal circumference. *Optimization*, 17:355–366, 1986.

[F.P79] F.P.Preparata. An optimal real time algorithm for planar convex hulls. *Communications of the ACM*, 22:402–405, 1979.

[GS90] M. Goodrich and J. Snoeyink. Stabbing parallel segments with a convex polygon. *Computer vision, Graphics and Image Processing*, 49:152–170, 1990.

[G.T90] G.Toussaint. Workshop on illuminating sets. Bellairs Research Inst. of McGill Univ., 1990.

[HDK64] H. Hadwiger, H. Debrunner, and V. Klee. *Combinatorial Geometry in the plane.* Holt, Rinehart and Winston, 1964.

[JMB92] S. Jadhav, A. Mukhopadhyay, and B. Bhattacharya. An optimal algorithm for the intersection radius of a set of convex polygons. In *Proc. of the Twelvth Annual FSTTCS Conference*, 1992.

[JS89] J.H.Reif and S.Sen. Polling: A new randomized sampling technique for computational geometry. In *Proc. of the Twenty-first Annual ACM Symposium an Theory of Computing*, May 1989.

[LMR] Kelly A. Lyons, Henk Meijer, and David Rappaport. Minimum polygon stabbers of isothetic line segments. Department of Computing and Information Science, Queen's University, Ontario, Canada.

[MTA86] M.Katchalski, T.Lewis, and A.Liu. Geometric permutations and common traversals. *Discrete & Computational Geometry*, 1:371–377, 1986.

[OL81] M. Overmars and H. Van Leeuwan. Maintenance of configurations in the plane. *Journal of Computer and System Sciences*, pages 166–204, 1981.

[O'R81] J. O'Rourke. An on-line algorithm for fitting straight lines between data ranges. *CACM*, 24:574–578, 1981.

[SM92] G. Srinivasaraghavan and Asish Mukhopadhyay. Stabbing simple planar sets. Technical Report TRCS-92-152, IIT Kanpur, Computer Science and Engg. Dept., 1992.

[Tam87] A. Tamir. Problem 4–2. Problems presented at the fourth NYU Computational Geometry Day, 1987.